Stability Analysis and Validation with OpenIPSL and Modelica Over-Current Relay Model

Manuel Navarro Catalan*
Department of Electrical Computer and Systems Engineering
Rensselaer Polytechnic Institute Troy, NY, USA
Email: *navarm2@rpi.edu

***Abstract*—Modeling and simulation in power systems are fundamental asset that all research and industry parties must explore. This paper also implements the stability analysis using the Modelica language and the OpenIPSL power systems library. The results of the instability studies are analyzed and validated to a previous Simulink based model. In addition, the Over Current Relay model previously presented is improved by making it OpenIPSL power systems library compatible.**

***Index Terms*—Over Current Relay, Modelica, OpenIPSL, Stability**

I. Introduction

One of the biggest threats to the equipment of a power system and the delivery of power to the customer are faults. Faults can be attributed to different events such ad component failures, misoperation, human error or even natural disasters. Relays are needed to detect and remove faults. Relays are crucial for the protection of power systems. They must be able to guarantee the safety of the personnel and the protection of the equipment, regardless of the situation. Therefore, these systems must be thoroughly designed in order to ensure the the last line of defense is able to provide the safety required. Modeling and simulation are crucial tools for the design and implementation of these systems.

The Modelica language has proved to be one of the most complete, and compatible modeling languages today. Its features include: open source, efficient equation-based for simple linearization, and multi domain modeling. In addition, Modelica is able to create portable models by being compatible with software such as EMTP-RV or EPHASORSIM. OpenIPSL is a Modelica language based open source power system component library. The library is being developed and maintained by the ALSETLab at Rensselaer Polytechnic Institute. Developing these stability scenarios using Modelica and OpenIPSL will open possibilities for further studies such as Hardware-In-The-Loop (HIL). They will be used to validate the results presented in [1] and [2]. In addition this study will be used to further stability understanding and the Over-Current relay (OCR) model presented in [3].

A. Related Work

There are three main works that work with this subject, the first is a presentation from Thiery Van Cutsem titled: ”Revisiting Phenomena and Instability Mechanisms” [1]. This presentation goes over the different types of instability scenarios (rotor angle, frequency and voltage), their definitions and the different features. The presentation also goes over the ”All-In-One” system (AIOS) and the outlines different test cases that were tested. Finally it provides the results of and expectations of the disturbances tested. This paper will serve as a base and guide for the re implementation of the AIOS The second paper titled: ”Detailed modelling, implementation and simulation of an all-in-one stability test system including power system protective devices” [2] gives a sample implementation using PowerFactory of the AIOS and a protective scheme developed for the AIOS. This will serve as an example of how to implement a protective scheme in the AIOS and will serve to validate the data with the OCR implementation. Lastly the work titled: ”Modeling of PMU-Based Automatic Re-synchronization Controls for DER Generators in Power Distribution Networks using Modelica and the OpenIPSL” [4] goes over load variation using stochastic theory. This work can be used in order to learn how to vary loads and truly test the efficiency of the protection scheme developed.

B. Contributions

The contributions of this paper are the following:

- Develop and produce an OpenIPSL compatible OCR model [3].
- To reproduce the AIOS test model presented in [1] using OpenIPSL [5].
- Validate stability tests performed in [1].
- Create a protection scheme based on the OCR model on the AIOS.

C. Paper Organization

The remainder of this paper is organized as follows. Section II will give an overview of an Over-Current relay (OCR) model [3]. Section III will provide insight of how the OCR model was linked to the OpenIPSL library [5]. Section IV will show how the ”All-In-One” test system was implemented in OpenIPSL. Section V depicts how stability analysis was performed in the ”All-In-One” system for different cases presented in [6]. Section VI will go over the implementation of records for the ’All-In-One” system. Section VII shows the discrepancies of the different cases presented. Section VIII shows an example of an OCR based protection scheme in the ”All-In -One” system. Section IX shows the results obtained in the study. Finally Section X describes the conclusion and future work to do.

II. Over-Current Relay Overview

In [3], an OCR Modelica model is presented. The model follows equation (1), in order to figure out when the trip signal will be issued once an over current event is detected. In this case, C and α are constants that will define the type of relay (standard, very, extremely or long inverse). I is the input current, I_s is the pick-up current and TMS is the time multiplier setting.

$$T = \frac{C}{\frac{I^{\alpha}}{I_s} - 1} TMS \quad (1)$$

Fig. 1. shows the Modelica model used to implement this OCR model. The red square is made up of the 3 protection algorithm components: $Timer$, $ExtractingTimeOfFault$, and $CalculatingOperatingTime$.

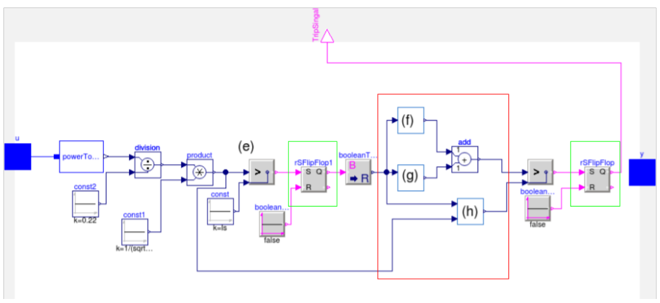


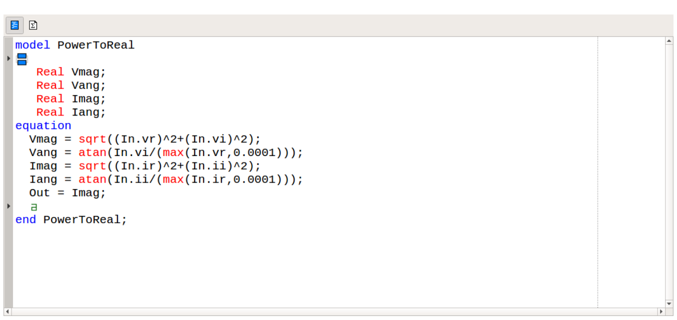
Fig. 1. Over-Current relay Modelica model.

III. Over-Curprent Relay - OpenIPSL Interconnection

One big drawback of the OCR model presented [3], is that the model was implemented in wave form, meaning that $real$ type connectors are designed to deal with time-varying signals. However, the OpenIPSL library has been developed through phasors. There were two main modification made: the $PowerToReal$ converter block and the implementation of latches.

A. $PowerToReal$ Converter Block

In order for the OCR to interconnect with phasors the $PowerToReal$ converter block was developed. Fig. 2 shows the code of the converter block.



```
model PowerToReal

  Real Vmag;
  Real Vang;
  Real Imag;
  Real Iang;
equation
  Vmag = sqrt((In.vr)^2+(In.vi)^2);
  Vang = atan(In.vi/(max(In.vr,0.0001)));
  Imag = sqrt((In.ir)^2+(In.ii)^2);
  Iang = atan(In.ii/(max(In.ir,0.0001)));
  Out = Imag;
  a
end PowerToReal;
```

Fig. 2. $PowerToReal$ converter block code.

The block is just composed of just two elements: a $Power$ input pin, and a $Real$ output pin. The block, will take in the real and imaginary part of both the voltage and current. Subsequently through the equation (2), the magnitude will be extracted. Where s is the real part of the vector and y is the imaginary part. The final output of the block will be magnitude of the current passing through the relay, which the relay will use to determine the appropriate protection action.

$$|A| = \sqrt{x^2 + y^2} \quad (2)$$

B. Latch Mechanisms

Due to the oscillating nature of magnitude signals, latches had to be implemented in order to ensure that once a fault was detected. If not implemented, in the case the relay detected a fault and then go back to the first state. This case can be see in Fig. 3. where the voltage drops after the fault time ($t = 1$ secs. If not latch, the relay would return to a "no fault state" and consequently oscillate between the "fault" and "no fault" states, interfering with the protection of the circuit.

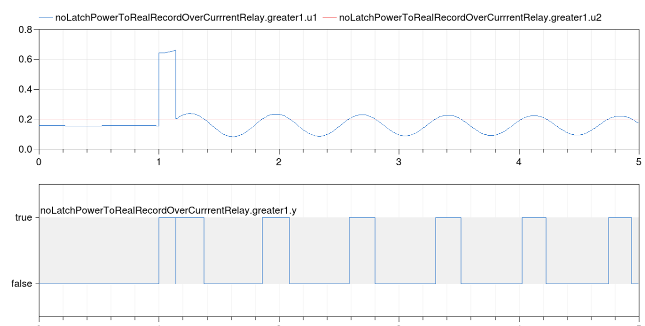


Fig. 3. No latch implementation oscillation between states.

The green circles in Fig. 1. show the two implementation of the latches. The latches comprise of two $rSFlipFlop$ blocks from the Modelica Standard Library (MSL). They are Boolean mechanisms that are: if $False$ then $False$, if $True$ then $True$. This will ensure that when a Boolean $True$ signal, meaning a fault happened, is issued it will stay as a "fault".

C. Interconnection Verifcation

In order to make sure that the modifications were correct, a sample faulted power system was created. As seen in Fig. 4. the power system was made up of a voltage source, two power lines, two buses and a PQ load.

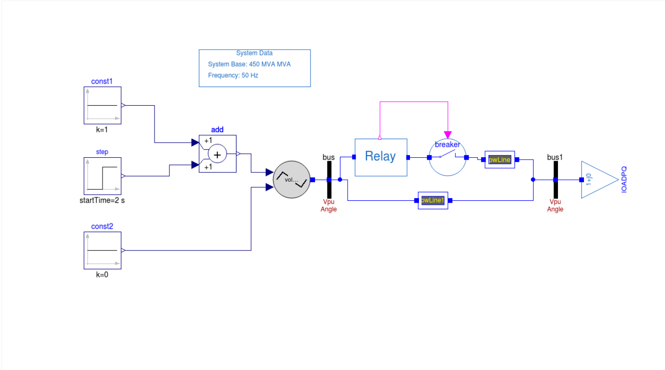


Fig. 4. OCR fault Trial power system.

A fault is applied at $t = 2$ secs. The OCR is set to standard inverse OCR parameters. As seen in Fig. 5., the OCR reacts properly to a fault, issuing a trip signal at $t = 2.68$ secs. Therefore, the modifications were successful, and the OCR can be porperly used with OpenIPSL.

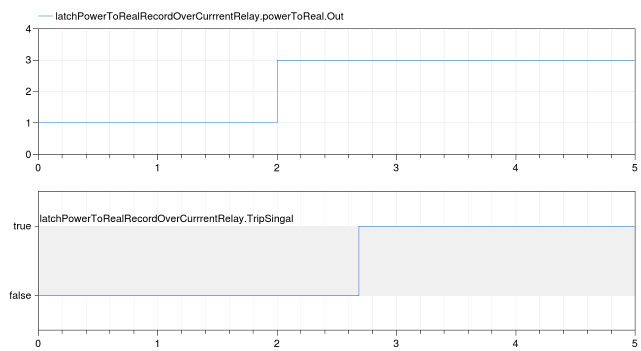


Fig. 5. Comparison between time of fault and trip signal.

IV. "All-In-One" Test System Re-Implementation

In order to perform the analysis tests, a simple system was developed, called the AIOS [1]. Different types of stabilities are tested (Transient, Frequency, Short-term voltage and Long-term voltage) through 8 different settings of the AIOS. The system had to be re implemented from Simulink to the Modelica language. Fig. 6. shows a one line diagram of the AIOS. In addition, the S_{base} = 750 MVA and the V_{base} = 380kV.

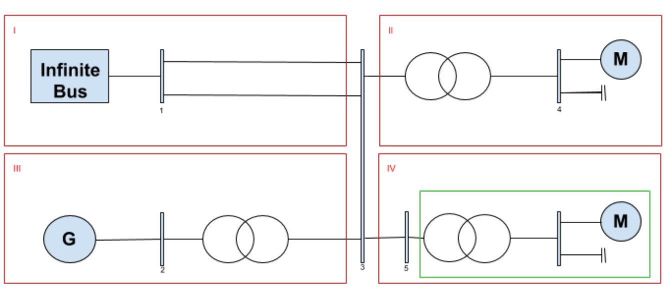


Fig. 6. "All-In-One" Test System.

As seen in Fig. 6., the system is made up of 4 areas:

- Area I - Transmission Lines.
- Area II - Transformer and Motor Load.
- Area III - Generator and Transformer.
- Area IV - Transmission Line and Load (Green Square).

A. System Re-Implementation Methodology

B. Area I

As seen in Fig. 7. Area I is made up of an Infinite bus (grid), and two 380kV transmission lines (Bus 2 $\rightarrow$ Bus 3).

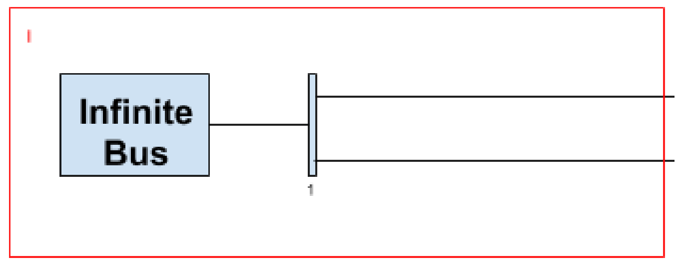


Fig. 7. All-In-One Test System Area I.

C. Area II

As seen in Fig. 8., Area II is made up of a 380kV/15kV transformer and a 750MVA, 15kV motor load.

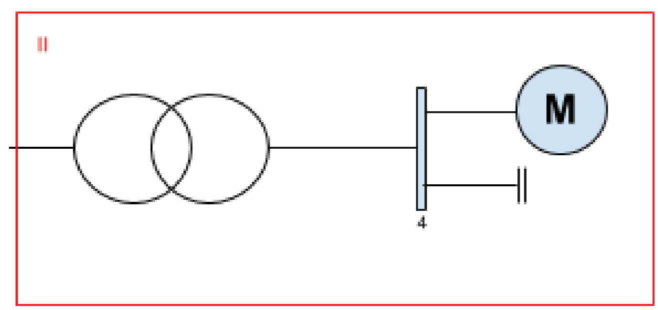


Fig. 8. All-In-One Test System Area II.

D. Area III

As seen in Fig. 9., Area III is made up of a 500MVA, 20kV 2^{nd} order motor generator and a 20kv/380kV transformer.

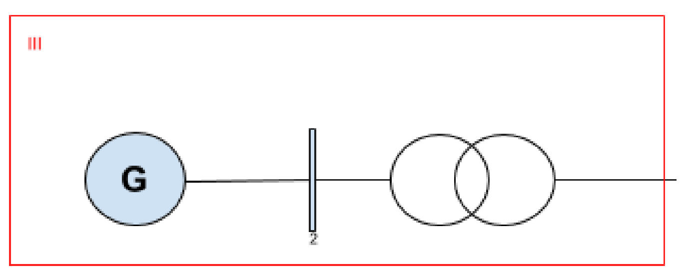


Fig. 9. All-In-One Test System Area III.

E. Area IV

As seen in Fig. 10., Area IV is made up of a 380kV transmission line and a variable PQ load.

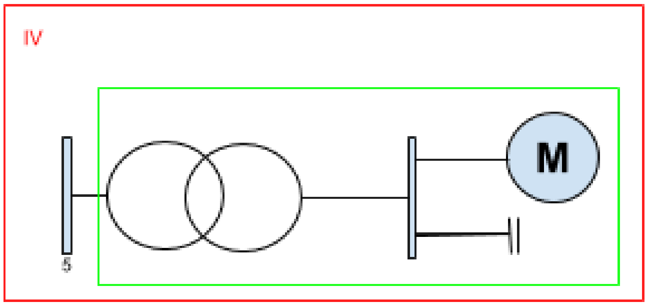


Fig. 10. All-In-One Test System Area IV.

The first step in the re implementation step was to take each of the components individually, and find/create the model and the parameters of it in OpenIPSL. This included comparing the given Simulink model with the PSAT [7] model for which the OpenIPSL library models were based on. The parameters of the different elements are the following (specifically for case $PF4$):

TABLE I
INFINITE BUS PARAMETERS.

Parameter	Value
V_b(kV)	380
V_0(pu)	1.04
$Angle_0(\circ)$	0

TABLE II
TRANSMISSION LINES (BUS 1 → BUS 3).

Parameter	Value
R(pu)	0
X(pu)	0.414
B(pu)	0
$t1$ (s)	∞
$t2$ (s)	∞
$opening$	1

TABLE III
380KV/15KV TRANSFORMER (BUS 3 → BUS 4).

Parameter	Value
S_b(MVA)	750
V_b(kV)	380
S_n(MVA)	750
V_n(kV)	380
rT (pu)	0
xT (pu)	0.08
m	1

TABLE IV
TYPE III MOTOR.

Parameter	Value
V_b(kV)	380
V_0 (pu)	1.04
$Angle_0(\circ)$	0
P_0(MW)	750
Q_0 (MVAR)	750
f_n(Hz)	50
Sup	1
R_s(pu)	0.031
X_s (pu)	0.01
R_{r1} (pu)	0.05
X_{r1}(pu)	0.07
X_m(pu)	3.20
H_m (s)	0.6
a (pu)	0.78
b(pu)	0
a (pu)	0
tup(s)	0

TABLE V
SHUNT CAPACITOR BANK.

Parameter	Value
Q_n(pu)	117
V_{base}(kV)	15
f_n (Hz)	50

TABLE VI
2_{nd} ORDER SYNCHRONOUS GENERATOR.

Parameter	Value
V_b(kV)	20
V_0 (pu)	1
$Angle_0(\circ)$	-9.4
P_0 (MW)	300
Q_0 (MVAR)	213
f_n(Hz)	50
S_n(MVA)	500
V_n(kV)	20
r_a(pu)	0
x_{1d}(pu)	.4148
M(s)	7
D	1

TABLE VII
20KV/380KV TRANSFORMER (BUS 2 → BUS 3).

Parameter	Value
S_b(MVA)	750
V_b(kV)	20
S_n(MVA)	500
V_n(kV)	20
rT (pu)	0
xT (pu)	0.08
m	$1/1.04$

TABLE VIII
TRANSMISSION LINES (BUS 3 → BUS 5).

Parameter	Value
R(pu)	0
X(pu)	0.03
B(pu)	0
$t1$ (s)	∞
$t2$ (s)	∞
$opening$	1

TABLE IX
PQ LOAD.

Parameter	Value
S_n(MVA)	500
V_b(kV)	380
V_0(pu)	1.0029
$Angle_0(\circ)$	0
P_0(MW)	500
Q_0(MVAR)	80
S_b(MVA)	750
f_n (Hz)	50

The re-implementation was done using a bottom-up approach, where each component was modeled individually, tested and then assembled in to the bigger system.

V. MODELING RESULTS

In order to keep the report brief, only 2 of the 8 different stability scenarios will be presented in this paper. The settings and results of all of the different scenarios are available in the appendix. The following are the results of Power Flow num. 1 and Power Flow num. 4. The angles of each of the buses and the P/Q consumed by the generator/motor/load will be the points of comparison in order to asses how well the model matches the theoretical result. For both cases (PF1 and PF 4), the results are nearly identical and prove that the Modelica model, matches with the theoretical AIOS model.

A. Scenario 1: Power Flow num. 1

This scenario takes into account the motor to be turned off (P = 0 MW, Q = 0 MVAR). The theoretical Power Flow results are shown in Fig. 11. and the Modelica Power Flow results are show in Fig. 12.

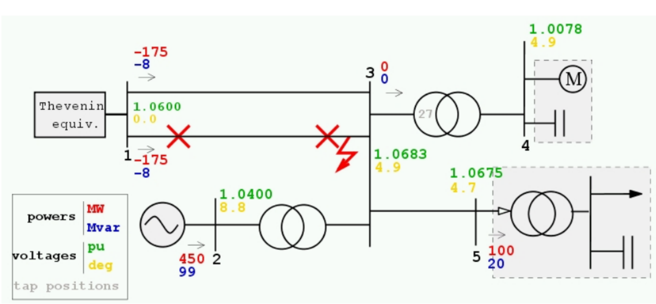


Fig. 11. Theoretical Power Flow Results PF 1 [1].

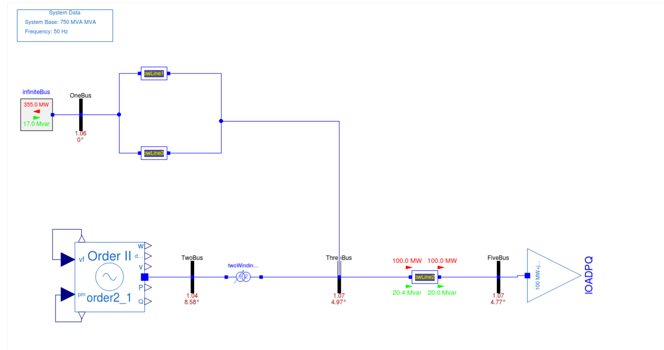


Fig. 12. Modelica Power Flow Results PF 1.

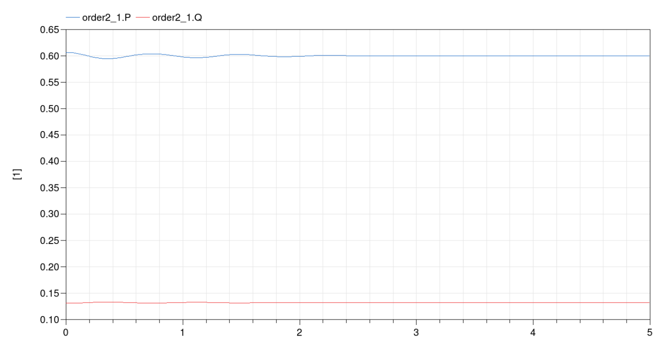


Fig. 13. Modelica Generator Power Flow Results PF 1.

As seen in Fig. 11., Fig. 12. and Fig. 13., the results are identical. Table X facilitates the comparison of values, and show the discrepancy in %.

TABLE X
POWER FLOW OF CASE 1.

Component	Theoretical Values	Modelica Values	Discrepancy%
B2 P(MW)	450	450	0.0
B2 V(pu)	1.04	1.04	0.0
B4 P(MW)	0	0	0.0
B4 Q(MVAR)	0	0	0.0
B5 P(MW)	100	100	0.0
B5 Q(MVAR)	20	20	0.0

B. Scenario 2: Power Flow num. 4

This scenario takes into account the motor to be turned off (P = 0 MW, Q = 0 MVAR). The theoretical Power Flow results are shown in Fig. 14. and the Modelica Power Flow results are show in Fig. 15.

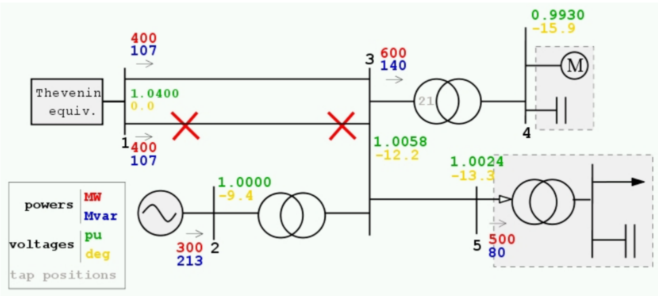


Fig. 14. Theoretical Power Flow Results PF 4 [1].

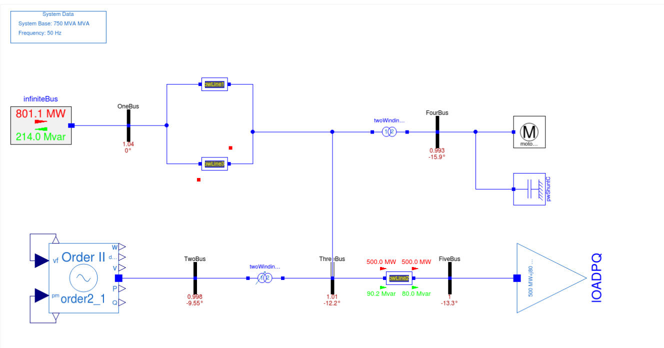


Fig. 15. Modelica Power Flow Results PF 4.

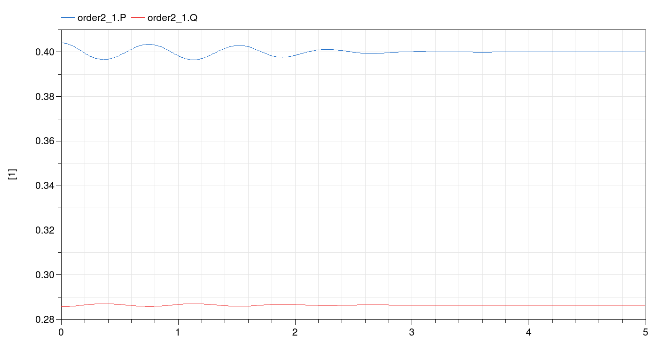


Fig. 16. Modelica Generator Power Flow Results PF 4.

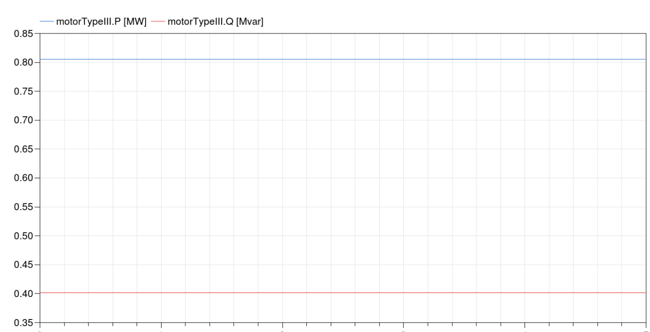


Fig. 17. Modelica Motor Power Flow Results PF 4.

As seen in Fig. 14., Fig. 15., Fig. 16 . and Fig. 17., the results are nearly identical. Table XI facilitates the comparison of values, and show the discrepancy in %. In this case the Generator has a .2% discrepancy.

TABLE XI
POWER FLOW OF CASE 4.

Component	Theoretical Values	Modelica Values	Discrepancy%
B2 P(MW)	300	300	0.0
B2 V(pu)	1.00	.998	0.2
B4 P(MW)	600	604.12	0
B4 Q(MVAR)	100	0	0
B5 P(MW)	500	500	0
B5 Q(MVAR)	80	80	0

VI. OVER-CURRENT RELAY BASED PROTECTION SCHEME

In [1], the protective scheme was based on setting the lines to open, at a given time after the known fault was detected. In this case, the new OpenIPSL compatible OCR model will be used to mitigate the faults. The $pwFault$ OpenIPSL component will be used in order to perform the line to ground faults on the system. Table XII shows the parameters of the fault. For the sake of consistency, the faulted cases analyzed and protected will be: PF 1 and PF 4. The OCR protection could be ans. The results will be compared to the theoretical results found in [1].

TABLE XII
LINE TO GROUND FAULT.

Parameter	Value
R(pu)	0.0
X(pu)	0.3
t_1(s)	2
t_2(s)	2.14

A. Scenario 1: Power Flow num. 1

Based on Fig. 18., a line to ground fault is applied for $t = 0.14$ secs. starting at $t = 1.00$ secs., the figure also shows the implementation of the fault and the OCR.

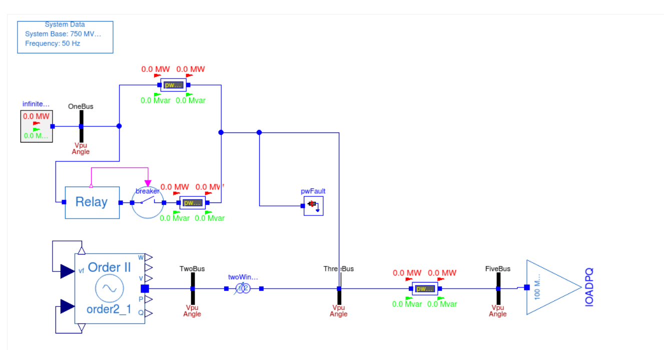


Fig. 18. Fault and OCR implementation in PF 1.

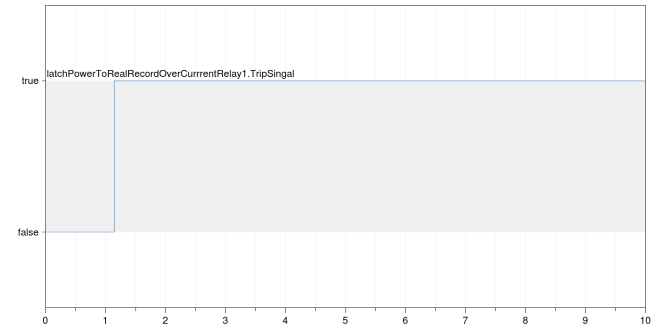


Fig. 19. Trip signal issued when fault is applied to PF 1.

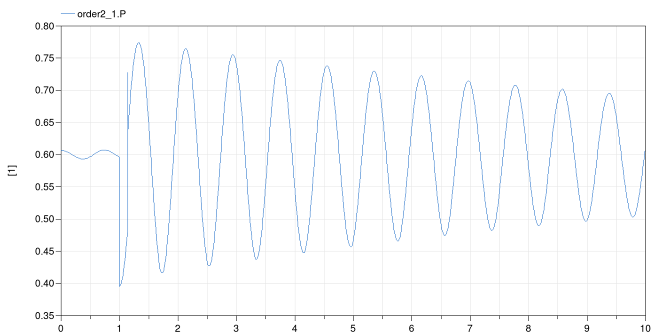


Fig. 20. Generator P during fault in PF 1.

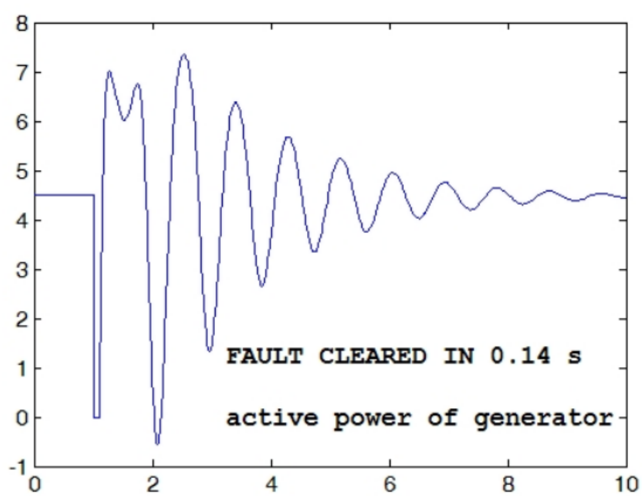


Fig. 21. Generator P during fault in PF 1 [1]

Fig. 19., shows how the trip signal was successfully issued at $t = 1.14$ secs. This confirms the successful incorporation and function of the OCR within an OpenIPSL model. Fig. 20. and Fig. 21. show the P of the generator during the fault. The Modelica model generator P, Fig. 19., has the same behaviour as the theoretical result expected, Fig. 21. However, the Modelica P signal does not damp as quickly as the theoretical model, due to the fact that the generator lacks an AVR, exciter, and governor, this will prevent the generator to damp after a fault. More over, the results show how the relay can be successfully used to develop a protection scheme that will act the same way as the theoretical model. However, the generator must first be modeled at a higher level and with the other components in order to obtain optimal results.

B. Scenario 2: Power Flow num. 4

Based on Fig. 22., a line to ground fault is applied for $t = 0.14$ secs. starting at $t = 1.00$ secs., the figure also shows the implementation of the fault and the OCR.This case is different since the motor load is no longer zero.

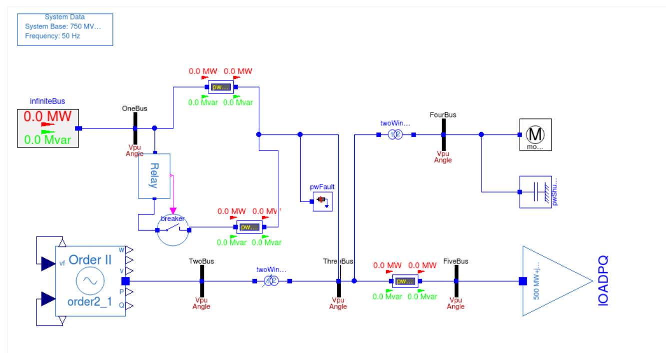


Fig. 22. Fault and OCR implementation in PF 4.

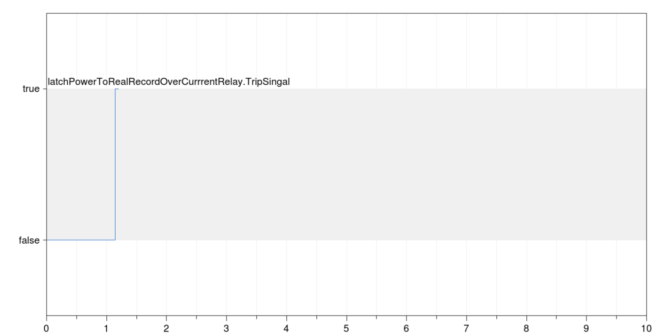


Fig. 23. Trip signal issued when fault is applied to PF 4.

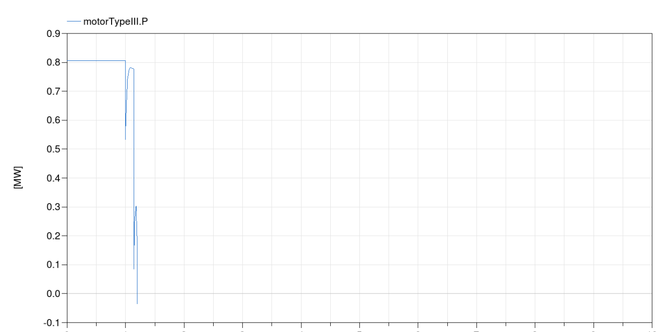


Fig. 24. Motor P during fault in PF 4.

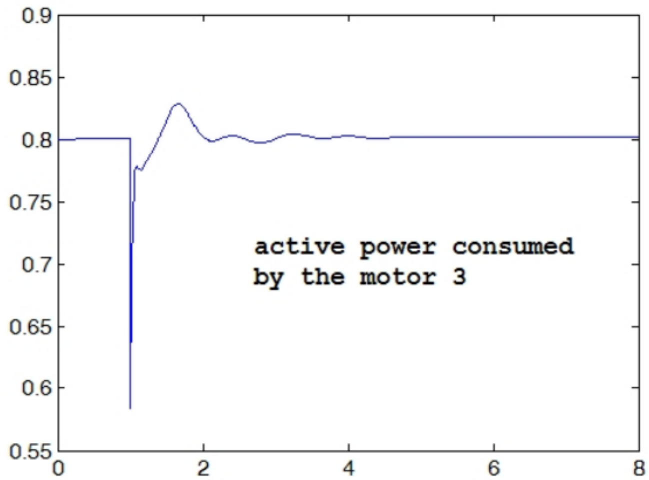


Fig. 25. Motor P during fault in PF 4 [1]

In this case, the simulation crashes shortly after the trip signal issue. This is due to the poor modeling of the motor and the generator, where the nonlinear system can no longer be solved once one of the circuits is tripping. However, as seen in Fig. 23., the trip signal is once again issue at the correct time $t = 1.14$ secs. Ideally, after the fault, the system would return to stability, as seen in Fig. 25. The discrepancies are due to the lack of detailed modeling of both the motor and generator. With a better modeling of both components, the protection scheme should output the same results. Regardless, both cases prove that the OCR model can be confidently used as a protection scheme in OpenIPSL and will perform as the theoretical results predict.

For both test cases, the Time Multiplier Setting (TMS) had to be changed in order to reach the desired $t = 0.14$ secs. tripping time. TMS was changed to $TMS = 0.21492$. In theory $0.1 \geq TMS \geq 1.0$. The OCR was kept as a standard inverse OCR, meaning $\alpha = 0.02$ and C $=$ $.14$. Another change made was that the pick up current was changed to: $\mathrm{I}_s = 0.73$.

VII. Conclusions and Future Work

The results of the study were very successful. The AIOS was re-implemented using Modelica and the OpenIPSL library.The OCR was successfully implemented and adapted in order the . The results obtained were in according to what was theoretically expected minus some discrepancies that can be attributed to the different simulation methods and the lower order generator and motor models. The main goals of the study were met.

There were a few shortcomings of this study. The first is the generator model. In this paper, a stand alone order 2 generator was presented. However, in the model presented in [2], the generator is made up of an AVR, Steam Turbine (order 6 generator), an Exciter and a Governor. The generator was made in such low complexity, due to the lack of a governor, exciter and AVR. However, priority was given to stability studies rather than the perfect modeling of the generator. The second is the lack of the LTC implementation. Due to time constraints, the LTC was not implemented, instead, the paper presented a model with a constant PQ load. This caused two test cases in [6] to be impossible to reproduce. Both shortcomings can be the reason to the discrepancies in power flow, and real/imaginary power dispatch. Another problem with the OCR model is the $ExtraactingTimeofFault$ block, where inside the integrator, the OCR will output the trip signal at the wrong time, unless the upper limit of the block is set to the fault time. This is unacceptable, since the relay should automatically detect this parameter. More time is needed, but the approach to solve this problem is to create a custom integrator block where the value of the upper limit of the integrator is set to the Boolean input of the block, this will ensure that the trip signal is issued at the appropriate time.

In future work, the generator would be modeled in the right order (6th order), an AVR, Exciter and Governor would be modeled in order to obtain a more accurate representation of the system. The motor model would be improved. Both would be done through parameter sweep methods in order to find the right parameters that would deliver the right dispatch. In addition, records should be implemented in order to achieve a single model that can switch easily between each of the test cases.

References

[1] C. D. Vournas, E. G. Potamianakis, C. Moors, and T. Van Cutsem, “An educational simulation tool for power system control and stability,” *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 48–55, 2004.

[2] R. Leelaruji and L. Vanfretti, “Detailed modelling, implementation and simulation of an all-in-one stability test system including power system protective devices,” *Simulation Modelling Practice and Theory*, vol. 23, pp. 36–59, 2012.

[3] M. Navarro Catalan and L. Vanfretti, “Over current relay modeling using modelica with cross-verification against a validated model,” in *7th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*. IEEE, 2019.

[4] B. Mukherjee and L. Vanfretti, “Modeling of pmu-based automatic re-synchronization controls for der generators in power distribution networks using modelica and the openipsl,” in *Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019*, no. 157. Linköping University Electronic Press, 2019.

[5] M. Baudette, M. Castro, T. Rabuzin, J. Lavenius, T. Bogodorova, and L. Vanfretti, “Openipsl: Open-instance power system libraryupdate 1.5 to itesla power systems library (ipsl): A modelica library for phasor time-domain simulations,” *SoftwareX*, vol. 7, pp. 34–36, 2018.

[6] “Revisiting Phenomena and Instability Mechanisms, note = Accessed: 2019-04-28.”

[7] L. Vanfretti and F. Milano, “Application of the psat, an open source software, for educational and research purposes,” in *2007 IEEE Power Engineering Society General Meeting*. IEEE, 2007, pp. 1–7.

APPENDIX

TABLE XIII
POWER FLOWS OF THE DIFFERENT TEST CASES.

Case Number	1	2	3	4	5	6	7
Bus 2 P(MW)	450	350	350	300	300	300	450
Bus 2 V(pu)	1.04	1.01	1.01	1.00	1.01	1.01	1.01
Bus 4 P(MW)	0	0	0	600	0	0	0
Bus 4 Q(MVAR)	0	0	0	100	0	0	0
Bus 5 P(MW)	100	400	500	500	1200	1500	1500
Bus 5 Q(MVAR)	20	80	80	80	0	150	150

The following sections show the different cases that were tested and show both the theoretical and Modelica results.

A. Power Flow Case 1; Result Comparison

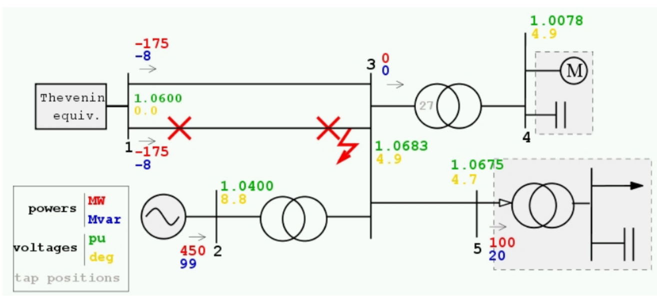


Fig. 26. Power Flow Case 1 [1].

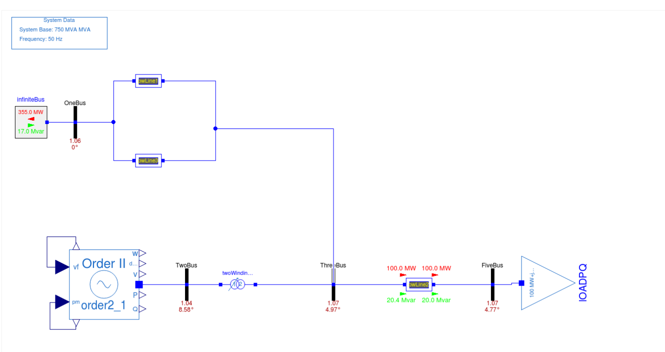


Fig. 27. Power Flow Case 1 Modelica.

B. Power Flow Case 2; Result Comparison

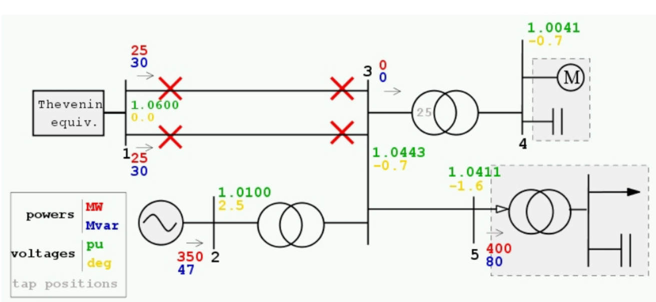


Fig. 29. Power Flow Case 2 [1].

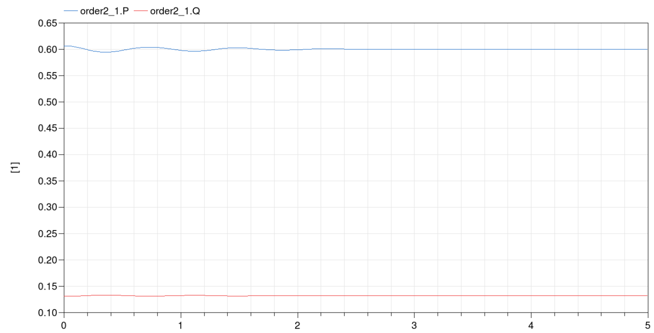


Fig. 28. Power Flow Case 1 Modelica (Generator).

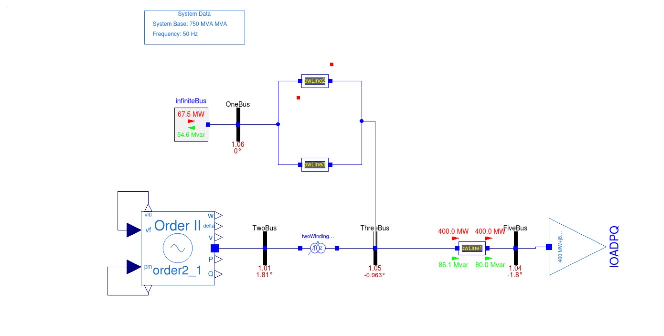


Fig. 30. Power Flow Case 2 Modelica.

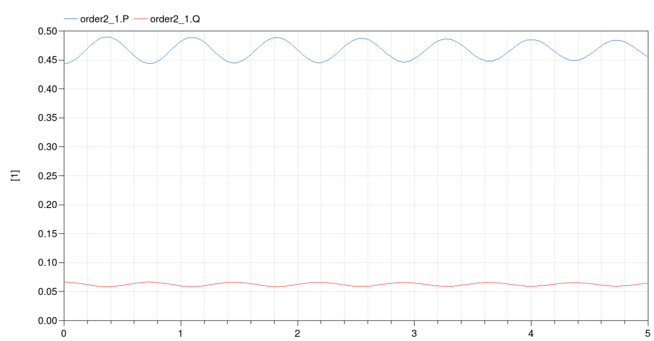


Fig. 31. Power Flow Case 2 Modelica (Generator).

C. Power Flow Case 3; Result Comparison

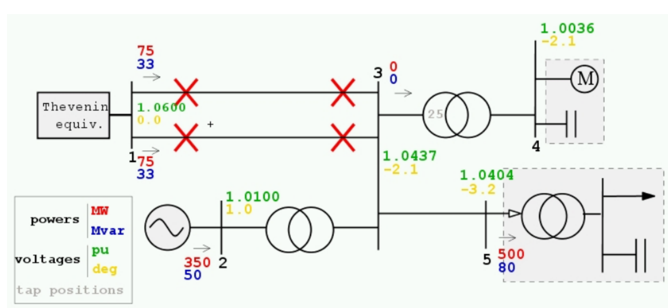


Fig. 32. Power Flow Case 3 [1].

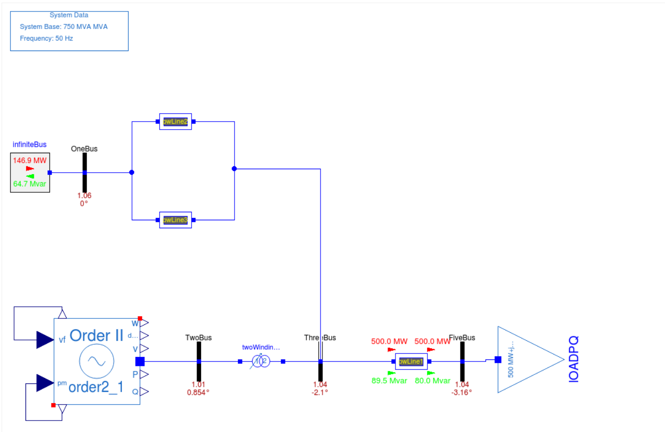


Fig. 33. Power Flow Case 3 Modelica.

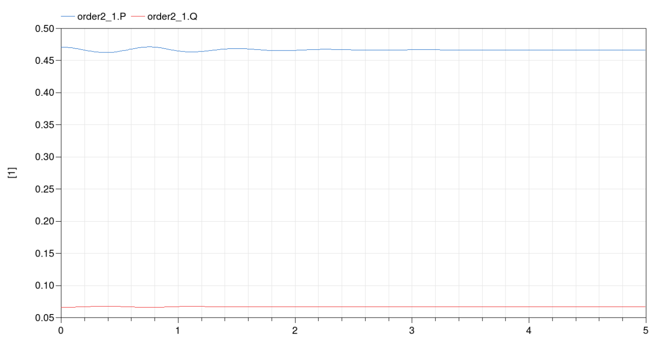


Fig. 34. Power Flow Case 3 Modelica (Generator).

D. Power Flow Case 4; Result Comparison

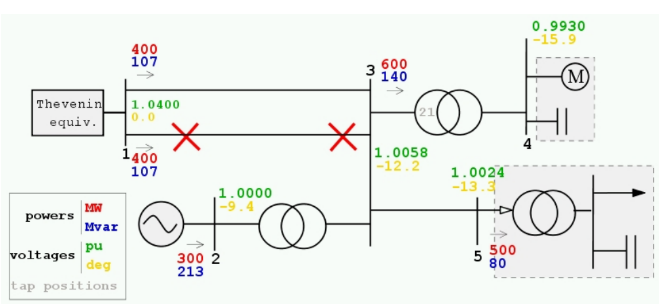


Fig. 35. Power Flow Case 4 [1].

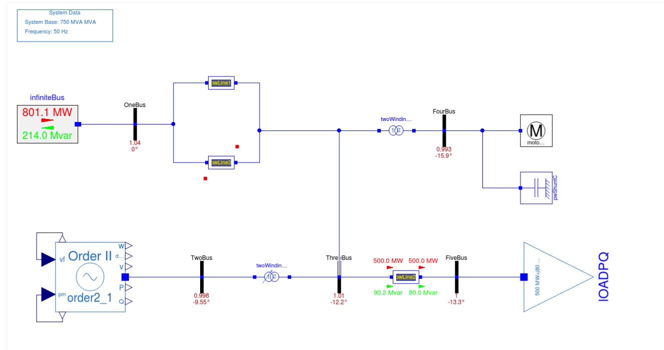


Fig. 36. Power Flow Case 4 Modelica.

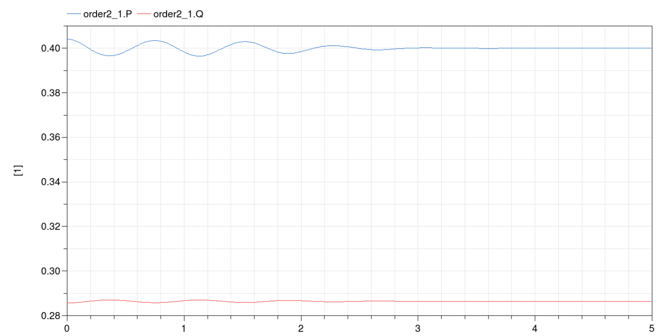


Fig. 37. Power Flow Case 4 Modelica (Generator).

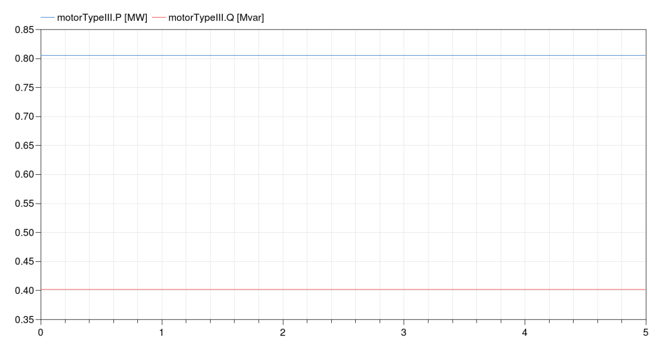


Fig. 38. Power Flow Case 4 Modelica (Motor).

E. Power Flow Case 5; Result Comparison

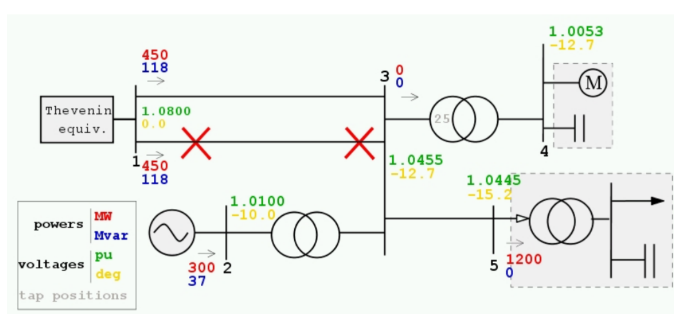


Fig. 39. Power Flow Case 5 [1].

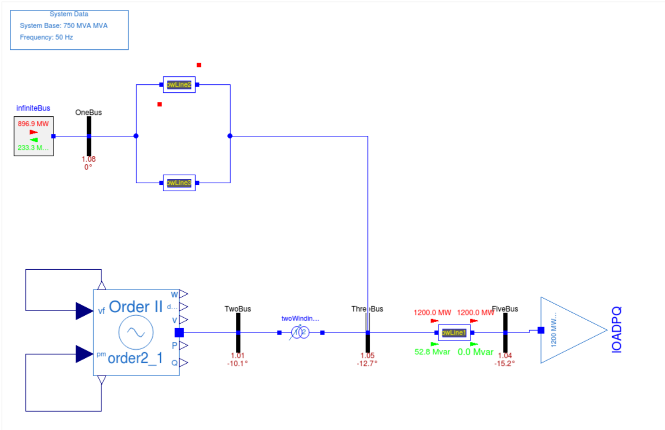


Fig. 40. Power Flow Case 5 Modelica.

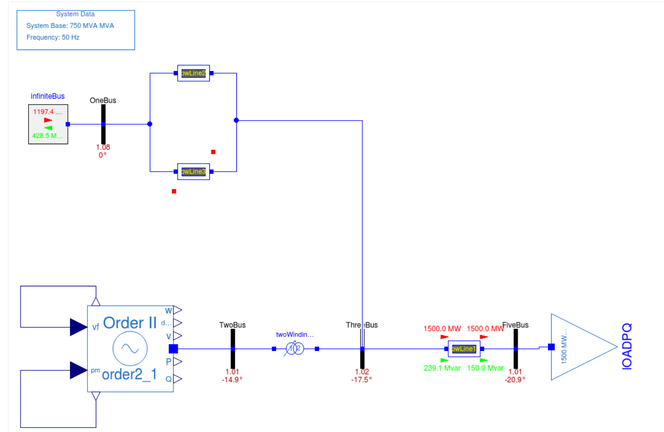


Fig. 43. Power Flow Case 6 Modelica.

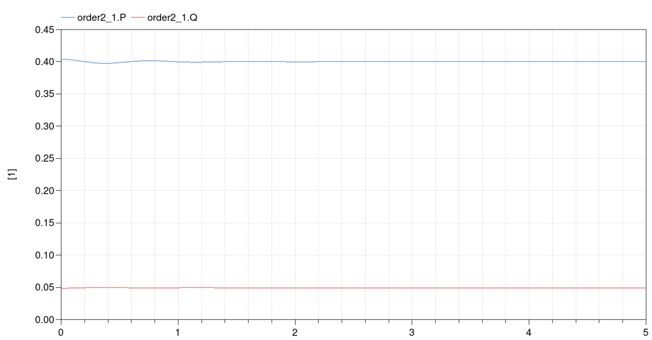


Fig. 41. Power Flow Case 5 Modelica (Generator).

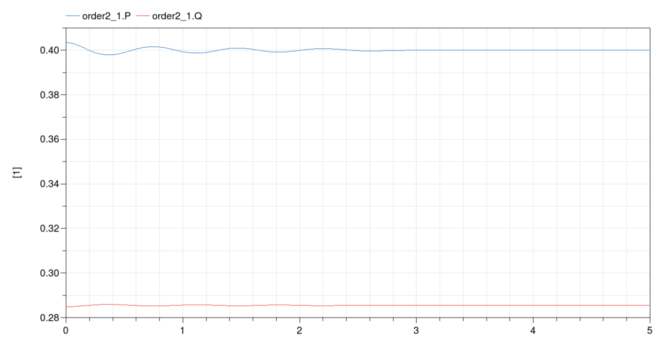


Fig. 44. Power Flow Case 6 Modelica (Generator).

F. *Power Flow Case 6; Result Comparison*

G. *Power Flow Case 7; Result Comparison*

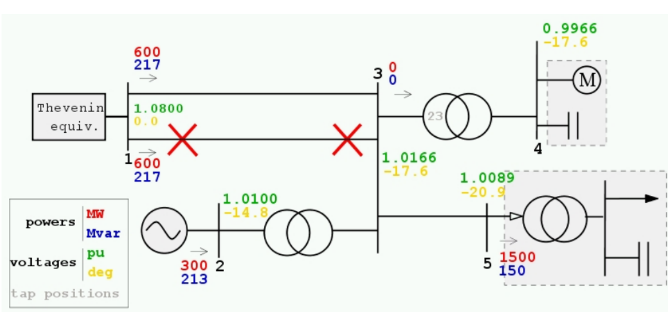


Fig. 42. Power Flow Case 6 [1].

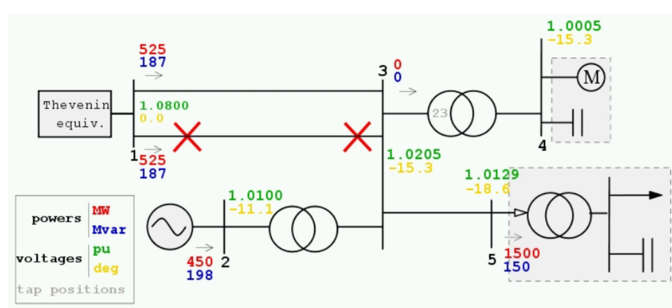


Fig. 45. Power Flow Case 7 [1].

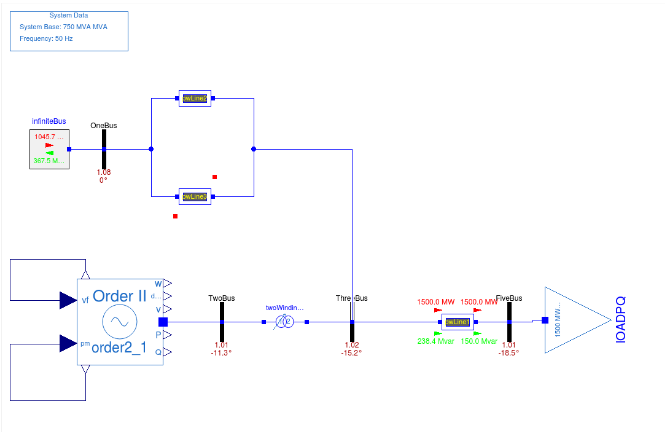


Fig. 46. Power Flow Case 7 Modelica.

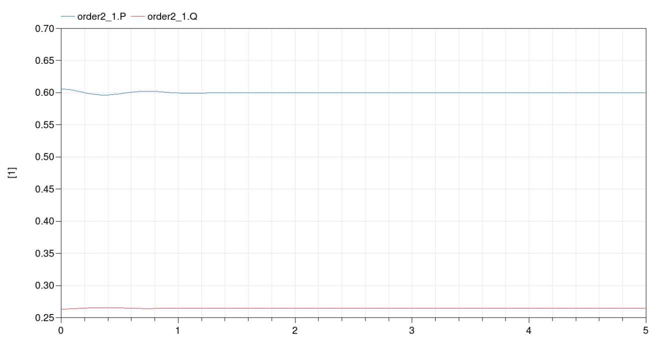


Fig. 47. Power Flow Case 7 Modelica (Generator).

H. Power Flow Case 8; Result Comparison

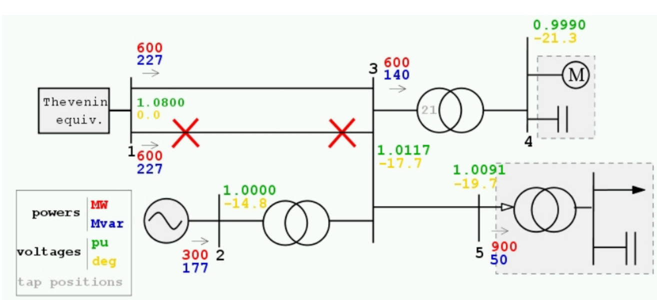


Fig. 48. Power Flow Case 8 [1].

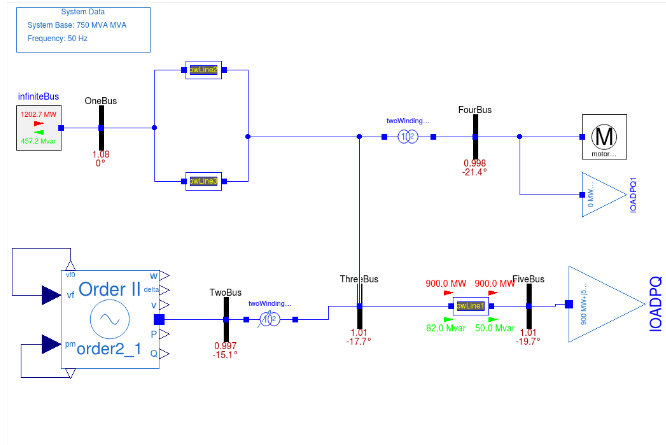


Fig. 49. Power Flow Case 8 Modelica.

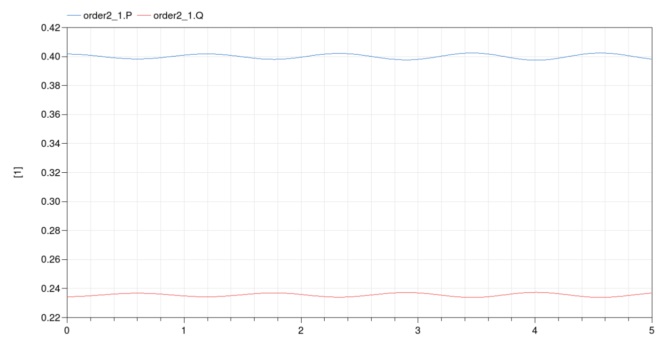


Fig. 50. Power Flow Case 8 Modelica (Generator).

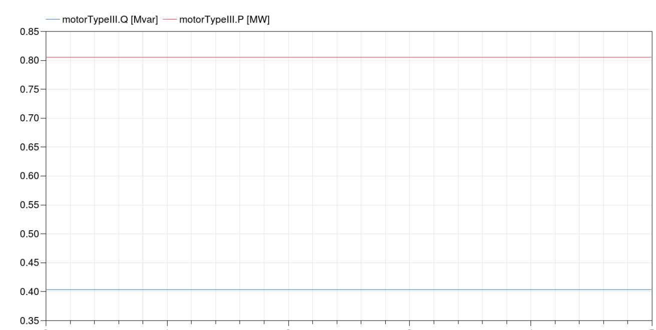


Fig. 51. Power Flow Case 8 Modelica (Motor).